Australian Journal of Basic and Applied Sciences, 9(11) May 2015, Pages: 822-826

ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

An Enhanced Instant-Fuzzy Search With Proximity Ranking For Video Retrieval

[1]C.S. Celin M.E. and [2]Dr. R. Nedunchelian

[1]*Department Of CSE, Saveetha University, Thandalam, Chennai-602105, India*
[2]*Professor, Department Of CSE,Saveetha University, Thandalam , Chennai-602105, India*

ARTICLE INFO

Article history:
Received 16 March 2015
Accepted 28 May 2015
Available online 22 June 2015

Keywords:
Data Mining, web mining, Fuzzy search, clustering.

ABSTRACT

The Instant is a search enhancement that shows results as you type. We are pushing the limits of our technology and infrastructure that help to get better search results, faster. Start typing and results appear right before. Until now, you had to type a full search term, hit return, and hope for the right results. Now results appear instantly as writing query, helping you see where you're headed, and every step of the way. Fuzzy search is the technique of finding strings that match a pattern approximately (rather than exactly). It is a type of search that will find matches even when users misspell words or enter in only partial words for the search. Responses for every query are supposed to answer with in milliseconds. With the help of Fuzzy Search Misspellings and types still provide relevant results and to increase the relevant score proximity of keyword is to be considered. In this paper, an approach that focuses on common phrases and develop an incremental-computation algorithm for efficiently segments a query into phrases to overcome the space and time limitations. This study is propose to index the video and develop an algorithm for efficiently retrieving the video that should meet high speed requirement on large data sets even there are too many answers from early-termination techniques such as proximity ranking. The high speed requirement should be achieved by creating a XML for the query and by crawler program. This proposal is conducted on real dataset of video and minimizes the tradeoffs between time, space and quality of the video.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: C.S. Celin M.E. and Dr. R. Nedunchelian., An Enhanced Instant-Fuzzy Search With Proximity Ranking For Video Retrieval. ***Aust. J. Basic & Appl. Sci.,*** *9(11): 822-826, 2015*

INTRODUCTION

Web mining is the application of data mining techniques to discover patterns from the Web. According to analysis targets; web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining. Web usage mining is the process of extracting useful information from server logs e.g. use Web usage mining is the process of finding out what users are looking for on the Internet. Some users might be looking at only textual data, whereas some others might be interested in multimedia data. Web Usage Mining is the application of data mining techniques to discover interesting usage patterns from Web data in order to understand and better serve the needs of Web-based applications. Usage data captures the identity or origin of Web users along with their browsing behavior at a Web site. Web usage mining itself can be classified further depending on the kind of usage data considered:

Web Server Data:

The user logs are collected by the Web server. Typical data includes IP address, page reference and access time.

A. ***Application Server Data:***

Commercial application servers have significant features to enable e-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.

B. **Application Level Data**:

New kinds of events can be defined in an application, and logging can be turned on for them thus generating histories of these specially defined events. It must be noted, however, that many end applications require a combination of one or more of the techniques applied in the categories above.

When used for research and investigation. Fuzzy searching is especially useful when researching unfamiliar, foreign-language, or sophisticated terms, the proper spellings of which are not widely known.

Corresponding Author: C.S. Celin M.E., 1Department Of CSE, Saveetha University, Thandalam, Chennai-602105, India
E-mail: celin.cs7@gmail.com

Fuzzy searching can also be used to locate individuals based on incomplete or partially inaccurate identifying information. Content data: are presented to the end-user appropriately structured. They can be simple text, images, or structured data, such as information retrieved from databases. Structure data: represent the way content is organized. They can be either data entities used within a Web page, such as HTML or XML tags, or data entities used to put a Web site together, such as hyperlinks connecting one page to another. Usage data: represent a Web site's usage, such as a visitor's IP address, time and date of access, complete path (files or directories) accessed, referrers address, and other attributes that can be included in a Web access log. User profile data: provide information about the users of a Web site. A user profile contains demographic information for each user of a Web site, as well as information about users' interests and preferences. Such information is acquired through registration forms or questionnaires, or can be inferred by analyzing Web usage logs.

I. Instant Search And Fuzzy Search:

The Instant search is an emerging information-retrievalparadigm in which a system finds answers to a query instantlywhile a user types in keywords character-by-character. Fuzzysearch further improves user search experiences by findingrelevant answers with keywords similar to query keywords. A main computational challenge in this paradigm is the high-speedrequirement, i.e., each query needs to be answered withinmilliseconds to achieve an instant response and a high querythroughput. At the same time, we also need good rankingfunctions that consider the proximity of keywords to computer relevance scores.In this paper, we study how to integrate proximity informationinto ranking in instant-fuzzy search while achieving efficient timeand space complexities. We adapt existing solutions on proximityranking to instant-fuzzy search. A naıve solution is computing allanswers then ranking them, but it cannot meet this high-speedrequirement on large data sets when there are too many answers,so there are studies of early-termination techniques to efficientlycompute relevant answers. To overcome the space and timelimitations of these solutions, we propose an approach that focuseson common phrases in the data and queries, assuming recordswith these phrases are ranked higher. We study how to indexthese phrases and develop an incremental-computation algorithmfor efficiently segmenting a query into phrases and computingrelevant answers. We conducted a thorough experimental studyon real data sets to show the tradeoffs between time, space, andquality of these solutions.

Instant Search:

As an emerging information-accessparadigm, instant search returns the answers immediatelybased on a partial query a user has typed in. For example, theInternet Movie Database, IMDB, has a search interface thatoffers instant results to users while they are typing queries1.When a user types in "sere", the system returns answerssuch as "Serena", "Serenity", "Serendipity", and
"Serena Williams". Many users prefer the experience ofseeing the search results instantly and formulating their queriesaccordingly instead of being left in the dark until they hitthe search button. Our recent study showed that this newinformation-retrieval method helps users find their answersquickly with less effort (Latha Shanmuga Vadivu, M., 2011).

Fuzzy Search:

Users often make typographical mistakesin their search queries. Meanwhile, small keyboards on mobiledevices, lack of caution, or limited knowledge about the datacan also cause mistakes. In this case we cannot find relevantanswers by finding records with keywords matching the queryexactly. This problem can be solved by supporting fuzzysearch, in which we find answers with keywords similar tothe query keywords

Related Work:

Find the documents containing a similar keyword by computing the union of the inverted lists of these similar keywords. For the last query keyword, we consider the union of the inverted lists for the completions of each prefix similar to it. We intersect these union lists to find all the candidate answers. Then we compute the score of each answer using a ranking function, sort them based on the score, and return the top-k answers.Accurately measuring the semantic similarity between words is an important problem in web mining, information retrieval, and natural language processing.In video retrieval, one of the main problems is to retrieve a set of video that is semantically related to a given user query.

Estimation of the semantic similarity between words or entities of Video using web search engines. Web search engines provide an efficient interface to this vast information. Page counts and snippets are two useful information sources provided by most web search engines. We present an automatically extracted lexical syntactic patterns-based approach to compute the semantic similarity between words or entities using text snippets retrieved from a web search engine and structured fuzzy search pattern in order to manage the video from search engine. This forces users to manually submit their queries to multiple search engines one after another until they find the information they need or give up their retrieval desire.

Australian Journal of Basic and Applied Sciences, 9(11) May 2015, Pages: 822-826

This CLDA algorithm is used for topic modeling which can improve the efficiency and performance to a significant extent.Determining neutral emotion also helps to sort out things for sellers in e commerce applications.

II. Proposed System:

In software, a module is a part of a program. Programs are composed of one or more independently developed modules that are not combined until the program is linked. A single module can contain one or several routines.

This Project consists of 5 Modules:

- Lexical Pattern Extraction
- Lexical Pattern Clustering
- Measuring Semantic Similarity
- Ranking search results

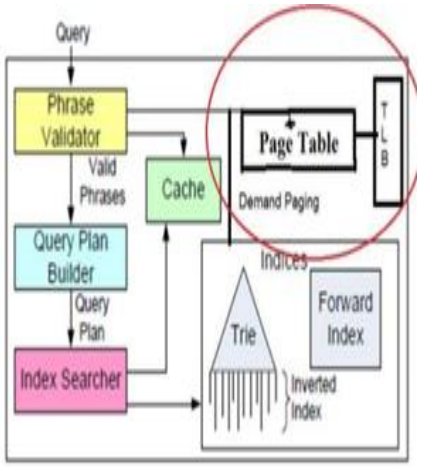

Fig. 1: Architecture Diagram for instant Fuzzy search.

Lexical Pattern Extraction:

In this module, Words in Page are extracted. It uses counts-based co-occurrence measures. Lexical Pattern Clustering. This can be problematic if one or both words are uncertainty, or when page counts are unreliable. On the other hand, the snippets returned by a search engine for the conjunctive query of two words provide useful clues related to the semantic relations that exist between two words. A snippet contains a window of text selected from a document that includes the queried words. Snippets are useful for search because, most of the time, a user can read the snippet and decide whether a particular search result is relevant, without even opening the url. Using snippets as contexts is also computationally efficient because it obviates the need to download the source documents from the web, which can be time consuming if a document is large.

2. Lexical Pattern Clustering:

Typically, a semantic relation can be expressed using more than one pattern. For example, consider the two distinct patterns, X is a Y, and X is a large Y. Both these patterns indicate that there exists and is-a relation between X and Y. Identifying the different patterns that express the same semantic relation enables us to represent the relation between two words accurately. According to the distributional hypothesis, words that occur in the same context have similar meanings. The distributional hypothesis has been used in various related tasks, such as identifying related words, and extracting paraphrases. If we consider the word pairs that satisfy (i.e., co-occur with) a particular lexical pattern as the context of that lexical pair, then from the distributional hypothesis, it follows that the lexical patterns which are similarly distributed over word pairs must be semantically similar.

3. Measuring Semantic Similarity:

It defined four co-occurrence measures using page counts. We showed how to extract clusters of lexical patterns from snippets to represent numerous semantic relations that exist between two words. In this module, we describe a machine learning approach to combine both page counts-based co-occurrence measures, and snippets-based lexical pattern clusters to construct a robust semantic similarity measure.

4. Ranking search results:

In this module, an automatic method to estimate the semantic similarity between words or entities using web search engines with ranking the search results occur. Accurately measuring the semantic similarity between words is an important problem in web mining, information retrieval, and natural language processing. Web mining applications such as, community extraction, relation detection, and entity disambiguation; require the ability to accurately measure the semantic similarity between concepts or entities. Based on the similarity between the user given search keyword, the ranking takes place.

The experiment results screenshots are shown in fig 2,3,4 below.

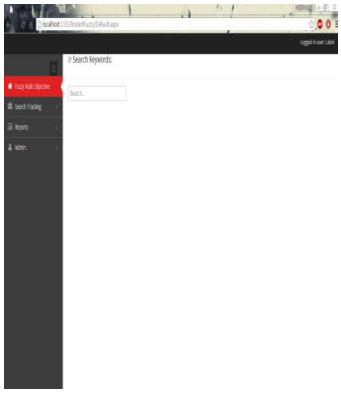

Fig. 2: Searching keywords in search box.

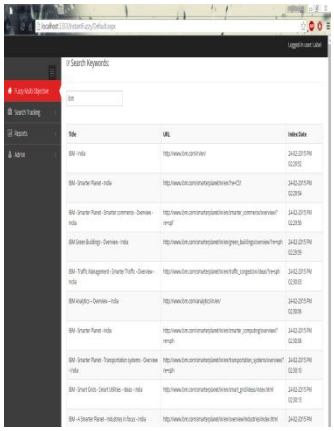

Fig. 3: Entering the Keyword IBM.

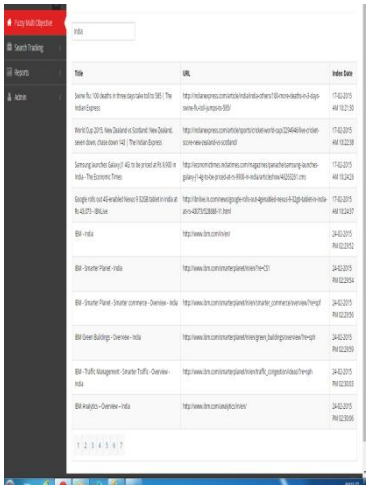

Fig. 3: Fuzzy refined Searching using keyword IBM india.

III. Conclusion:

The analyses that focuses on common phrases and develop an incremental-computation algorithm for efficiently segments a query into phrases to overcome the space and time limitations. This study is propose to index the video and develop an algorithm for efficiently retrieving the video that should meet high speed requirement on large data sets even there are too many answers from early-termination techniques such as proximity ranking. The high speed requirement should be achieved by creating a XML for the query and by crawler

program. This proposal is conducted on real dataset of video and minimizes the tradeoffs between time, space and quality of the video.

REFERENCES

Latha Shanmuga Vadivu, M., Raja ram and S.N. Sivanandam, 2011."A Survey On Semantic Web Mining Based Web Search Engines''.

Prakash, S., Raghavendra, Shreya Roy Chowdhury, SrilekhaVedula kameswari, 2011. "Web Usage Mining using Statistical Classifiers and Fuzzy Artificial Neural Networks''.

Poongothai, M., Parimala and Dr.S.Sathiyabama, 2011. "Efficient Web Usage Mining with Clustering''.

Haeed, R., Aghabozorgi, The YeangWah, 2011. "Using Incremental Fuzzy Clustering to Web Usage Mining''**.**

Sch. of Comput. & Inf. Technol., Beijing Jiaotong Univ., Beijing, China, Hong Shen, Yingpeng Sang ; Hui Tian, " Preserving Ranked Fuzzy Keyword Search over Encrypted Data" .

Dongsheng Wang; Coll. of Comput., Nat. Univ. of Defense Technol., Changsha, China, Shaojing Fu,Ming Xu , 2010. "A Privacy-Preserving Fuzzy Keyword Search Scheme over Encrypted Data ''.

Frigui, H., 2012. Dept. of Electr. &Comput. Eng., Univ. of Memphis, TN, USA "Simultaneous categorization of text documents and identification of cluster-dependent keywords''.

Chee Sheen Chan, 2012. Sch. of Comput. & IT, Taylor's Univ., Subang Jaya, Malaysia ;Johar, A. ; Jer Lang Hong, "Contextual information for image retrieval systems".

Imin ren, 2013. Electron. & Inf. Eng. Dept., Tianjin Inst. Of Urban Constr., Tianjin, China, "Research of Web Data Mining Based on Fuzzy Logic and Neural Networks''.

Xu choujan, Data Min. Lab., Nanjing Normal Univ., Nanjing, China, 2011. "Design and Implementation of Basic Educational Web Resources Gathering System".

Bast, H. and I. Weber, 2006. "Type less, find more: fast autocompletion searchwith a succinct index," in *SIGIR*, 364–371.

Bast, H., A. Chitea, F.M. Suchanek and I. Weber, 2007. "Ester: efficient search on text, entities, and relations," in *SIGIR*, 671–678.

Bast, H. and I. Weber, 2007. "The completesearch engine: Interactive, efficient,and towards ir& db integration," in *CIDR*, 88–95.

Ji, S., G. Li, C. Li and J. Feng, 2009. "Efficient interactive fuzzy keyword search," in *WWW*, 371–380.

Chaudhuri, S. and R. Kaushik, 2009. "Extending autocompletion to tolerateerrors," in *SIGMOD Conference*, 707–718.

Li, G., S. Ji, C. Li and J. Feng, 2009. "Efficient type-ahead search on relationaldata: a tastier approach," in *SIGMOD Conference*, 695–706.

Hadjieleftheriou, M. and C. Li, 2009. "Efficient approximate search on stringcollections," *PVLDB*, 2(2): 1660–1661.

Chakrabarti, K., S. Chaudhuri, V. Ganti and D. Xin, 2008. "An efficient filterfor approximate membership checking," in *SIGMOD Conference*, 805–818.

Chaudhuri, S., V. Ganti and R. Motwani, 2005. "Robust identification of fuzzyduplicates," in *ICDE*, 865–876.

Behm, A., S. Ji, C. Li and J. Lu, 2009. "Space-constrained gram-based indexing for efficient approximate string search," in *ICDE*, 604–615.

Fagin, R., A. Lotem and M. Naor, 2001. "Optimal aggregation algorithms formiddleware," in *PODS*.

Zhang, F., S. Shi, H. Yan and J.R. Wen, 2010. "Revisiting globally sorted indexes for efficient document retrieval," in *WSDM*, 371–380.